

Cahier de spécification

Projet The Independent Supervisor

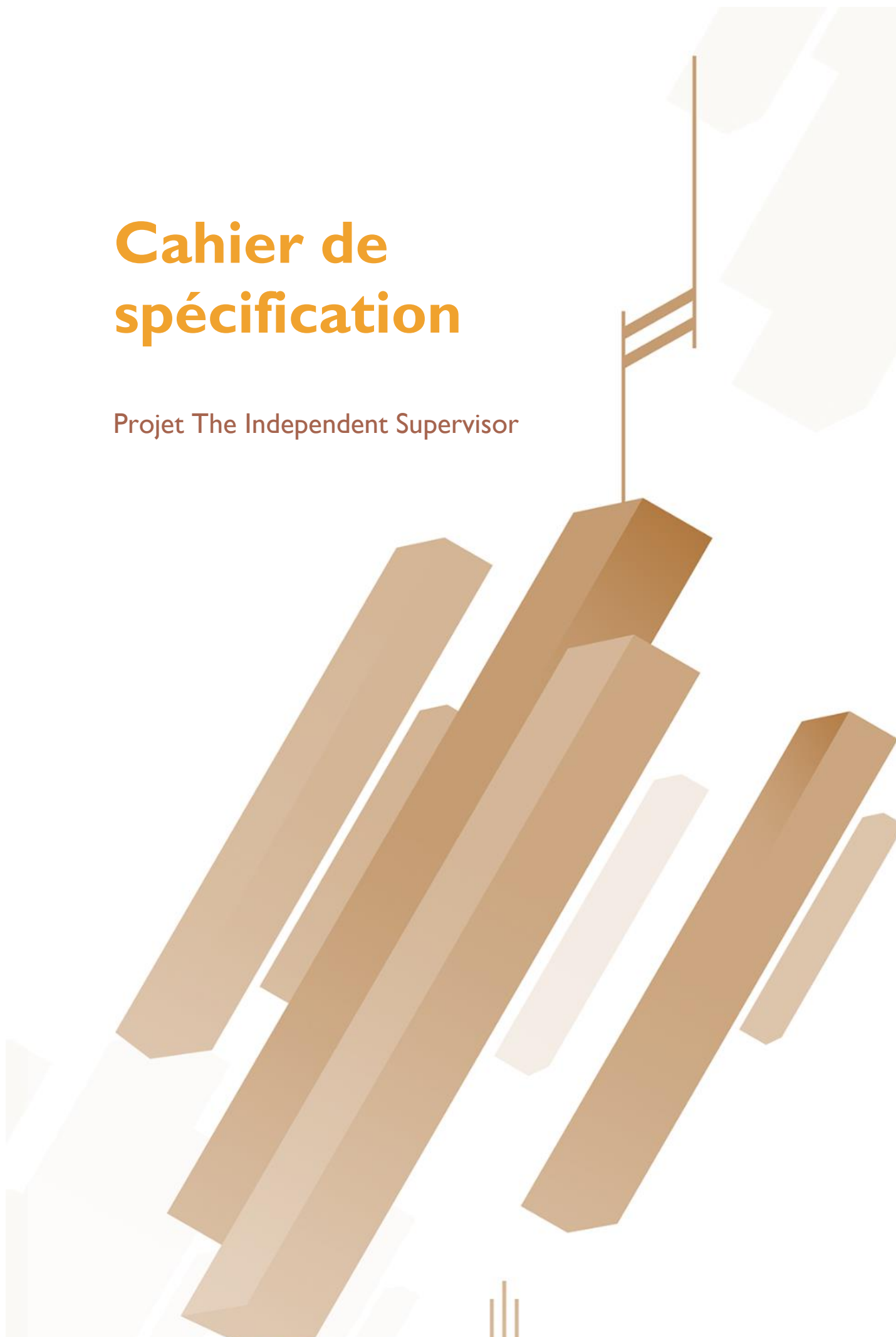


Table des matières

I. Cahier de spécification système	4
1. Introduction	4
2. Contexte de la réalisation	5
a) Contexte	5
b) Objectifs	6
c) Hypothèses	6
d) Bases méthodologiques	6
3. Description générale	7
a) Environnement du projet	7
b) Caractéristiques des utilisateurs	7
c) Fonctionnalités et structure générale du système	8
d) Contraintes de développement, d'exploitation et de maintenance	9
4. Description des interfaces externes du logiciel	10
a) Interfaces homme/machine	10
b) Interfaces logiciel/logiciel	10
5. Architecture générale du système	11
6. Description des fonctionnalités	12
a) Définition de la fonction « Gestion d'utilisateur »	12
b) Définition de la fonction « Gestion du parc »	13
c) Définition de la fonction « Consultation »	14
d) Définition de la fonction « Maintenance & Supervision »	15
e) Définition de la fonction « Communication »	16
7. Conditions de fonctionnement	17
a) Performances	17
b) Capacités	17
c) Modes de fonctionnement	17
d) Contrôlabilité	17
e) Sécurité	17
f) Conformité aux standards	17
II. Plan de développement	18
1. Découpage du projet en tâches	18
a) Tâche « Cahier de spécifications »	18
b) Tâche « Cahier d'analyse »	19
c) Tâche « Étude & Mise en place »	20

d) Tâche « Réalisation de l'application Web »	21
2. Planning	22

I. Cahier de spécification système

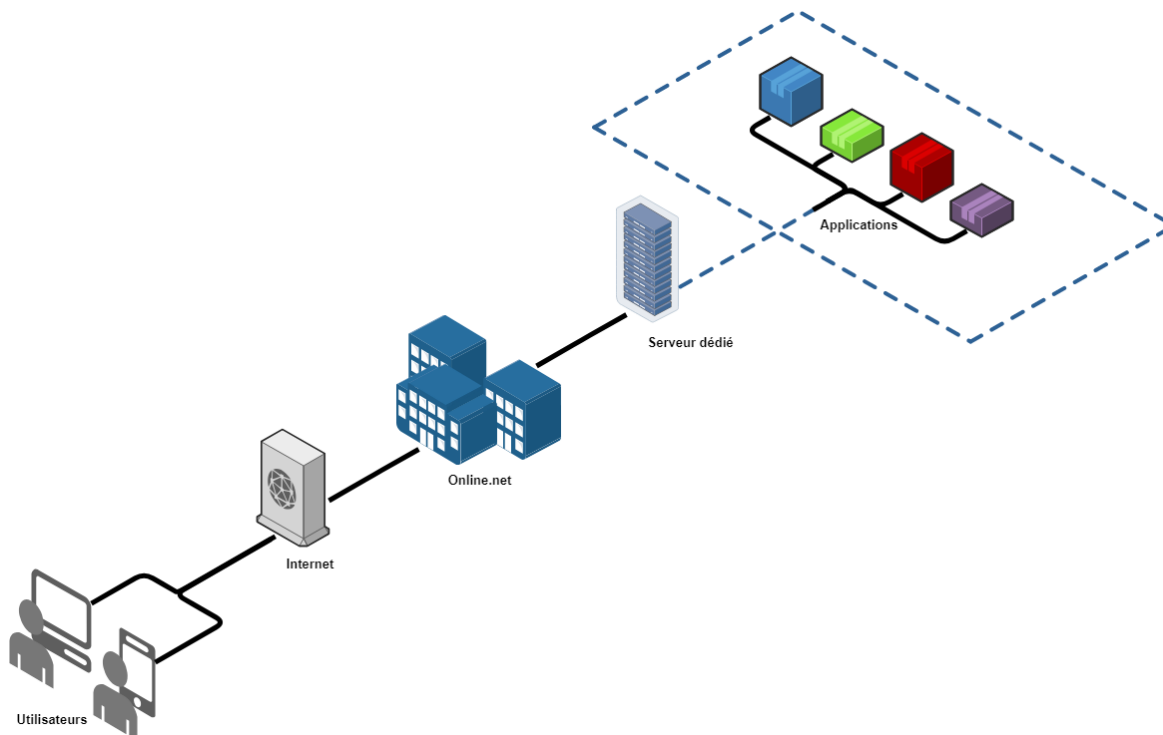
1. Introduction

The Independent Gamers est une équipe de joueurs amateurs, cofondé en 2013 par Brandon SIMON-VERMOT, cette équipe partage l'idée de fournir une ambiance divertissante en proposant diverses activités et la possibilité de rencontrer des joueurs réguliers sur une panoplie de jeux divers. The Independent Supervisor est le fruit d'une idée venant du chef d'équipe de l'équipe de joueurs amateur qui possède un serveur dédié, sur lequel plusieurs serveurs de jeux sont installés.

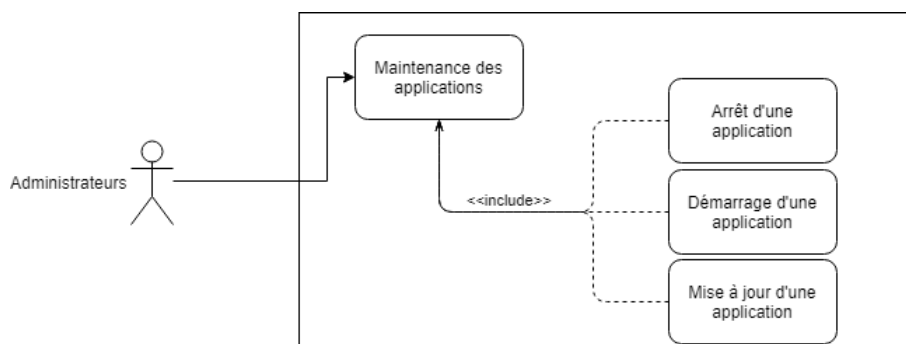
- Brandon SIMON-VERMOT : Chef de projet & Client
- Aneur SOUKHAL : Encadrant de projet
- Membres de l'équipe The Independent Gamers : Utilisateurs principaux

2. Contexte de la réalisation

a) Contexte



Chaque serveur de jeu, chaque application, présent sur le serveur dédié, qui est actuellement sous Ubuntu, a donc besoin d'être maintenu, d'être redémarré s'il y a une erreur, via des lignes de commandes. Cela demande du temps et un accès au serveur via une application de télémaintenance ou via un hyperviseur, actuellement KVM.



L'intérêt de ce projet, est de superviser les serveurs de jeux et offrir la possibilité de les mettre à jour automatiquement, de les maintenir. Ce projet part de zéro, aucun site Web et les outils liés (serveur Web, BDD, ...) ne sont déjà installés sur le serveur dédié, seuls les serveurs de jeux sont actuellement présents. Cette application web permettra donc de pouvoir exécuter des commandes grâce aux informations enregistrer par les administrateurs de l'application. Notamment les informations autour de chaque serveur de jeu (emplacement du dossier du serveur, commande de lancement, commande de sauvegarde).

b) Objectifs

Les objectifs du projet seront de fournir une application Web « from scratch », ainsi que tous les outils liés (serveur Web, BDD, ...), avec ces fonctionnalités présentes :

- Gestion du parc d'application (ajout, suppression d'un serveur de jeux)
- Superviser des serveurs de jeux (redémarrage auto. en cas de crash)
- Maintenir les serveurs de jeux (start/stop/update)
- Prévenir la maintenance d'un serveur (Flux RSS, Mail, Twitter, ...)
- Gestion d'utilisateurs (Administrateurs, Utilisateurs)
- Consultation des performances du serveur dédié
- Gestion des ressources du serveur dédié (arrêt d'applications si besoin)

c) Hypothèses

Il est nécessaire au projet d'avoir un accès à Internet, afin de prendre le contrôle du serveur dédié et pouvoir déployer le projet. Il n'est cependant pas nécessaire au développement du projet, puisqu'il est possible de développer un projet web sur n'importe quel ordinateur muni d'un éditeur de texte. Cet accès Internet est nécessaire pour accéder à l'application depuis n'importe quelle plateforme.

d) Bases méthodologiques

Dans le cadre de ce projet, des langages axés développement Web seront utilisés avec une méthode de développement adaptatifs, dans le but d'obtenir un livrable utilisable depuis n'importe quel navigateur depuis n'importe quelle plateforme avec des résolutions variable (Windows, Linux, Mac, Android, iOS, ...).

3. Description générale

a) Environnement du projet

Les applications cibles sont compatibles avec un système d'exploitation Linux principalement, plusieurs systèmes d'exploitation sont disponibles. Le livrable du projet devra être déployable sur le serveur dédié fourni, dont voici les caractéristiques :

Le serveur dédié est composé de :

- ✓ Host : **Online.net**
- ✓ CPU : **Intel® Xeon® D-1531** (6 cores, 12 threads, 2.2GHz)
- ✓ RAM : 32GB
- ✓ Stockage : 2 x 250GB SSD
- ✓ Server distributions : Ubuntu, Debian, CentOS, Windows, Archlinux, FreeBSD, Fedora
- ✓ Virtualization distributions : ESXi, Proxmox, CoreOS
- ✓ Panel distributions : cPanel
- ✓ Desktop distributions : Xubuntu, Lubuntu

b) Caractéristiques des utilisateurs

Dans cette application web, il y aura deux niveaux d'utilisateur :

- Utilisateur basique
- Administrateur

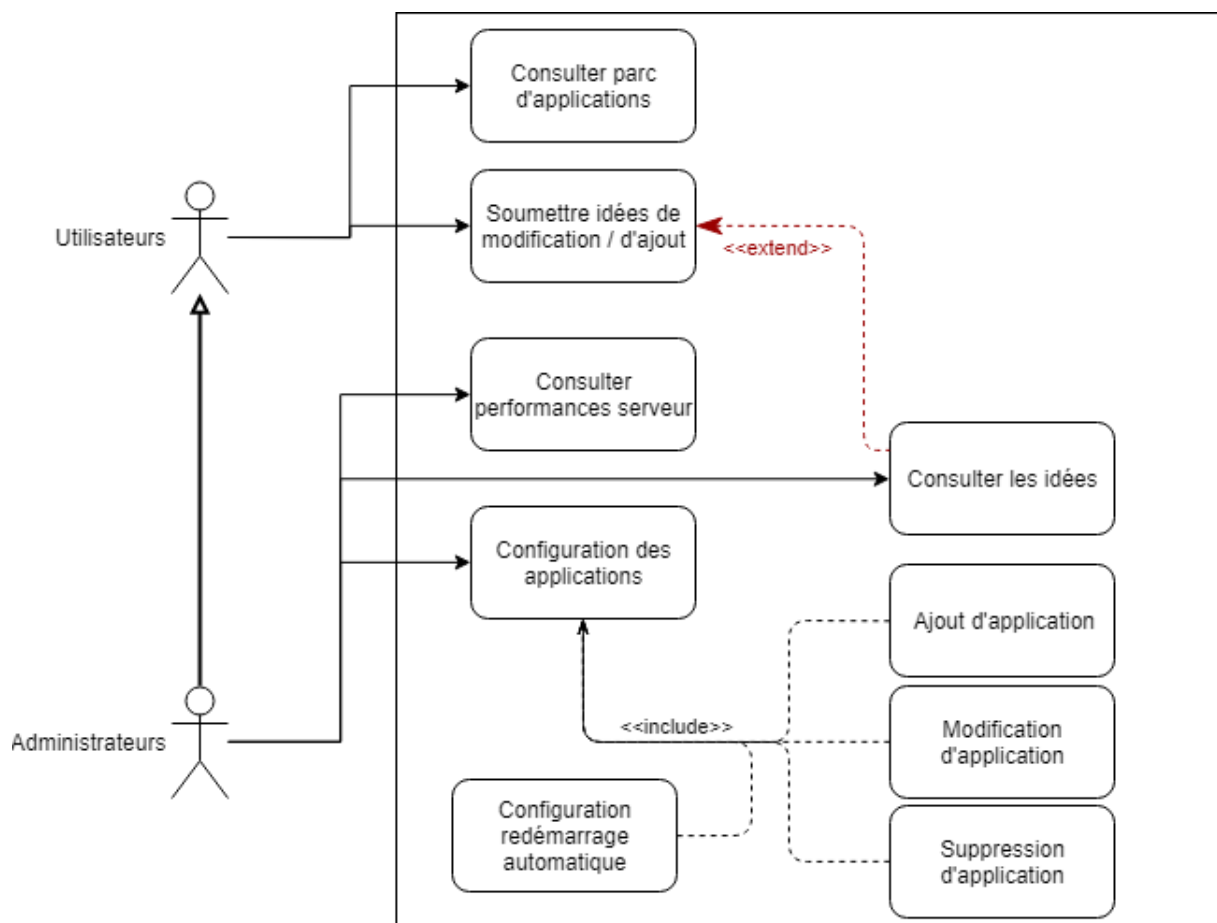
Les utilisateurs (basiques et administrateurs) de l'application sont des membres de l'équipe de joueurs, ils possèdent donc un minimum de connaissance de l'informatique, avec des niveaux variables.

Les utilisateurs basiques auront accès en lecture seule au parc d'application, ce qui permettra donc de consulter l'état des serveurs. Ils peuvent cependant accéder à une partie communication avec les administrateurs pour soumettre des idées de modification ou d'ajout d'une application.

Les administrateurs pourront quant à eux, pouvoir modifier depuis l'application web, le parc d'application (ajout, modification, suppression d'une application), configurer le redémarrage automatique des applications, ainsi que gérer les utilisateurs et leurs droits.

c) Fonctionnalités et structure générale du système

Peu importe l'utilisation et le niveau d'utilisateur, une authentification est nécessaire au préalable. Ci-dessous, on peut retrouver le diagramme de cas d'utilisation général :



d) Contraintes de développement, d'exploitation et de maintenance

Contraintes de développement

Il y a peu de contraintes de développement, puisque la plupart des ordinateurs actuels dispose d'un éditeur de texte et la possibilité d'obtenir (via une connexion Internet par exemple) un serveur web contenant le projet (ex : WAMP, LAMP, ...), ce qui est le strict minimum pour développer l'application web. Le projet peut être développé en local afin d'obtenir un « serveur de test » et le serveur dédié serait le « serveur de production ». Cependant une contrainte de développement est présente pour rendre l'application « responsive », afin de pouvoir afficher cette application correctement sur n'importe quelle navigateur web, sur n'importe quelle plateforme.

Maintenance et évolution du système

Ce projet fait partie de plus grande envergure, il est donc nécessaire d'obtenir une application web adaptative, afin de rendre le développement d'un autre projet plus facile, cela marche aussi pour des fonctionnalités qui peuvent être implémentés par la suite.

4. Description des interfaces externes du logiciel

a) Interfaces homme/machine

Pour notre interface homme/machine, qui sera affiché sur n'importe quel navigateur web et n'importe quelle plateforme, il sera important que notre utilisateur puisse retrouver rapidement et facilement ce qu'il cherche. Les alertes seront-elles aussi simplifiées pour l'utilisateur, pour comprendre un minimum l'information rencontrées, qu'elle soit une erreur ou non. Il est donc important qu'il y ait un maximum de choses transparentes pour l'utilisateur et que ce ne soit pas une tâche pénible, par exemple, afin qu'il puisse remonter facilement le moindre souci rencontré sur l'une des applications du parc.

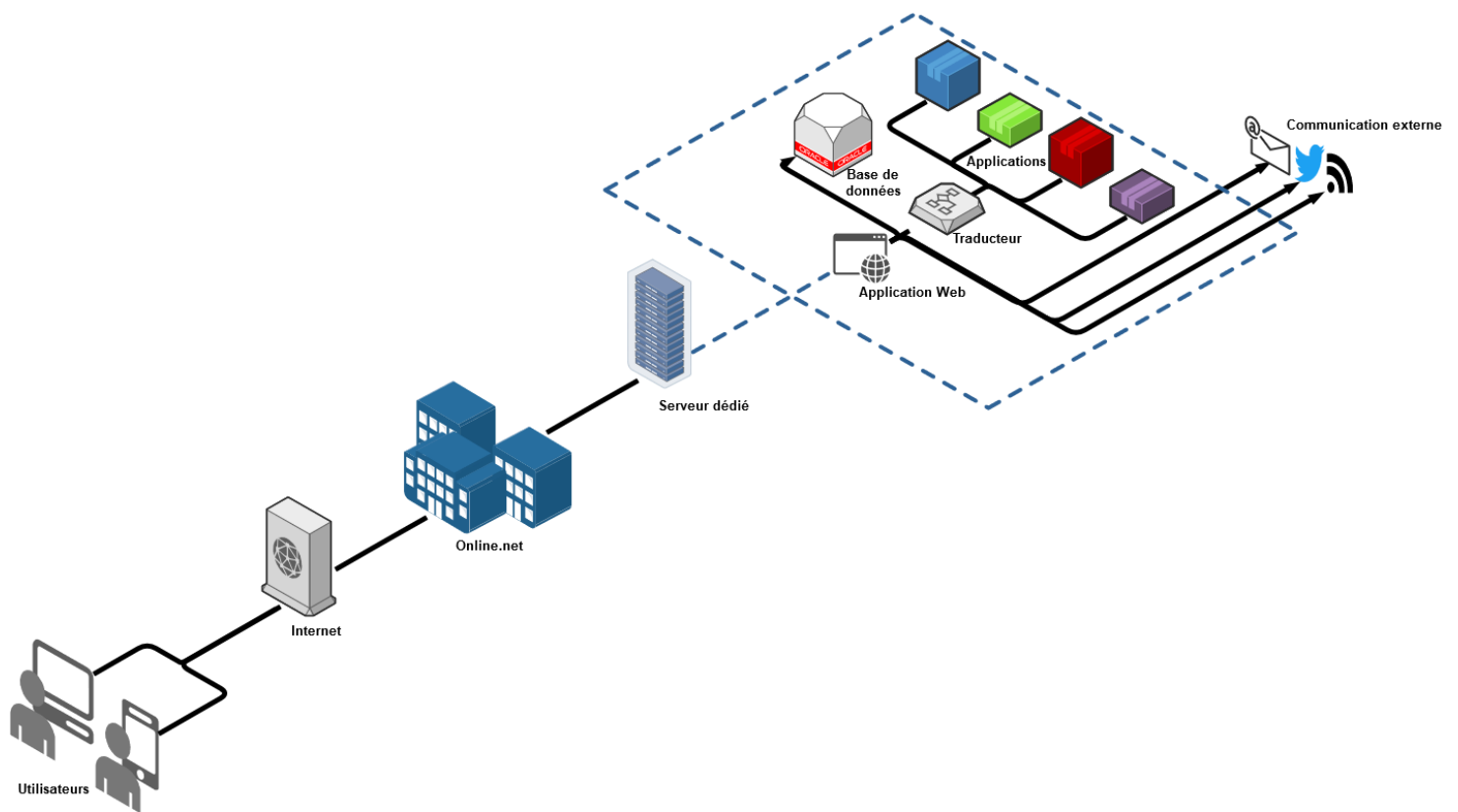
Une aide guidant notre utilisateur sera envisagée afin que peu importe le niveau en informatique de notre utilisateur, il sera utilisé toutes les fonctionnalités de l'application web.

b) Interfaces logiciel/logiciel

Pour une application web fonctionnel, nous avons besoin d'une base de données. L'accès se fera directement via l'application. Elle sera quasiment toujours sollicitée, puisque nous l'utiliserons dans ces différentes actions :

- Affichage du parc d'applications, de leurs statuts et de leurs configurations
- Enregistrement des idées postés par les utilisateurs
- Configuration de chaque application
- ...

5. Architecture générale du système



Pour mieux représenter le projet, ci-dessus se trouve un schéma comprenant les différents acteurs et éléments constituant l'architecture générale du système envisageable à la fin du projet. Dans un premier temps nos acteurs seront nos utilisateurs (administrateurs ou non), ils devront avoir une connexion Internet pour accéder à l'application Web, cette application est hébergée sur un serveur dédié appartenant au fournisseur Online.net.

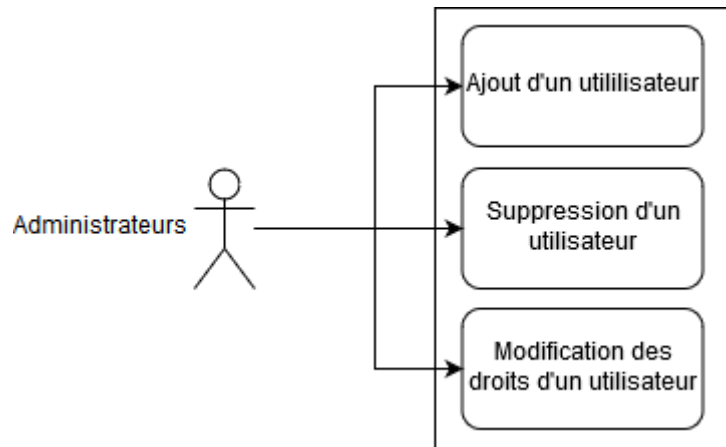
L'application Web étant le cœur du projet, elle est associée à différents éléments, une base de données contenant les différentes configurations de nos applications, dans le cadre de ce projet ce seront des serveurs de jeux. Ces applications sont différentes ce qui indique qu'elles peuvent avoir un moyen de communication différent, d'où l'intérêt d'avoir un élément qui nous servira de traducteur, dans le but de transmettre correctement l'information souhaitée.

Cette même application Web permettra de communiquer des informations sur l'état actuel des serveurs de jeux, pour cela il devra être possible de communiquer via différents moyens afin d'en informer nos utilisateurs, via par exemple le réseau social Twitter ou encore un flux RSS.

L'application Web devra pouvoir envoyer des mails aux administrateurs pour les prévenir en cas de problème mais aussi pour toute demandes et suggestions envoyés par nos utilisateurs via l'application Web.

6. Description des fonctionnalités

a) Définition de la fonction « Gestion d'utilisateur »



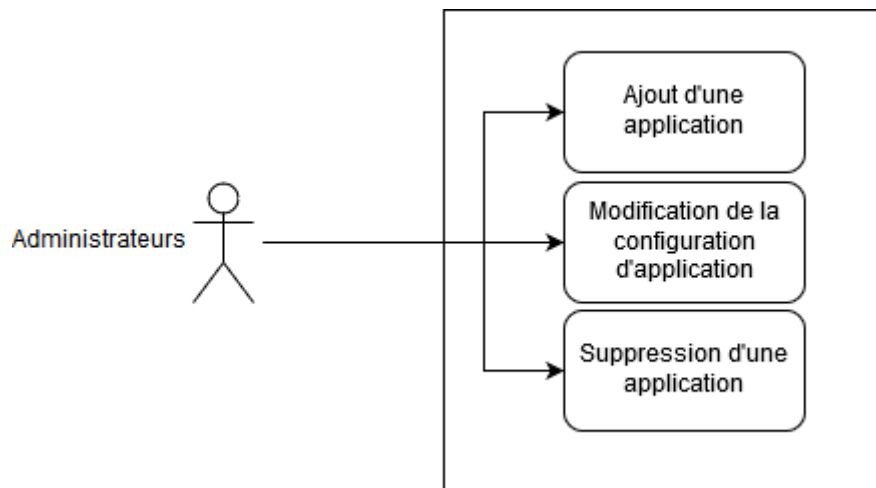
Identification de la fonction « Gestion d'utilisateur »

La fonction « Gestion d'utilisateur » permettra d'ajouter ou éditer les droits d'un utilisateur afin d'accéder à l'application Web. Il sera possible de supprimer un utilisateur. Cette fonction fait partie des fonctions facultative. Puisqu'elle n'est pas obligatoire au bon fonctionnement du projet, c'est un plus pour nos utilisateurs mais pas indispensable. Cependant, les fonctions « Gestion du parc » et « Maintenance & Supervision » doivent rester inaccessible pour les visiteurs.

Description de la fonction « Gestion d'utilisateur »

« Gestion d'utilisateur » est une fonction permettant d'attribuer des droits aux différents utilisateurs qui seront créé et supprimés via cette même fonction. Ces droits permettront aux utilisateurs d'avoir accès à certains services. Il y a au minimum deux rôles présents, les utilisateurs basiques et les administrateurs. Les utilisateurs basiques obtiennent un rôle en « lecture seule », ils ne pourront pas utiliser de fonctions modifiant le parc d'applications présent sur le serveur dédié. Contrairement aux administrateurs qui pourront eux utiliser ces fonctions.

b) Définition de la fonction « Gestion du parc »



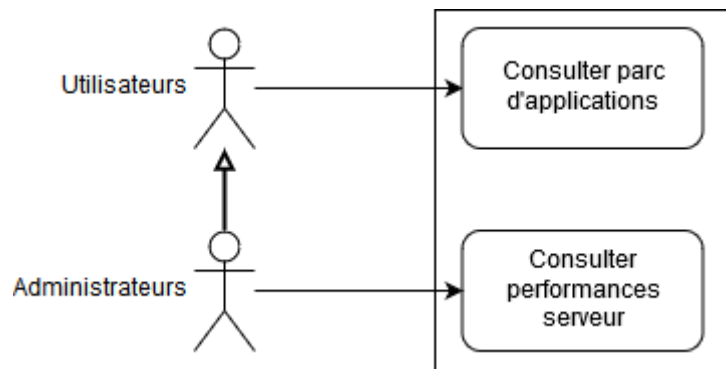
Identification de la fonction « Gestion du parc »

La fonction « Gestion du parc » est liée à la gestion de notre parc d'applications, ce qui signifie qu'elle sera semblable à la fonction « Gestion d'utilisateur » mais on pourra ajouter et éditer la configuration d'une application et il sera possible d'en supprimer une. Cette fonction est primordiale au projet, puisque c'est une partie importante de l'application Web.

Description de la fonction « Gestion du parc »

Cette fonction permettra d'ajouter une application à notre parc, ou supprimer une application déjà existante. Ce parc d'applications correspond aux applications qui seront superviser par notre application Web. Il est possible d'éditer la configuration liée à une application, cela permet de modifier au minimum la commande de lancement permettant le démarrage de l'application, la commande permettant la mise à jour de l'application et la commande permettant l'arrêt de l'application. Il est donc possible de retrouver d'autres informations qui peuvent différer selon les applications.

c) Définition de la fonction « Consultation »



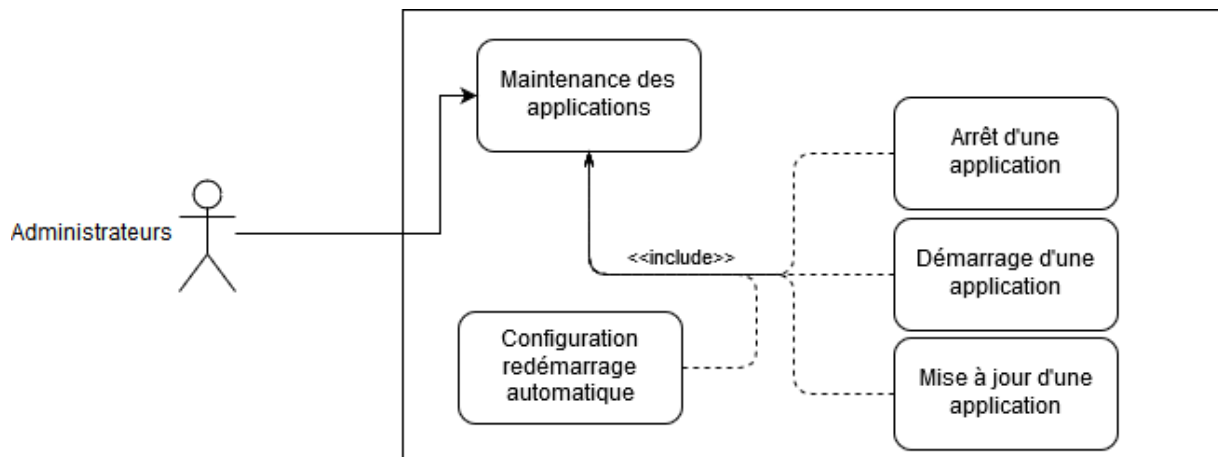
Identification de la fonction « Consultation »

Une fonction de « Consultation » permettra d’afficher les informations liées à chaque application de notre parc ou les performances en temps réel du serveur dédié. Cette fonction est secondaire au projet, la consultation n’est pas nécessaire au bon fonctionnement de l’application Web cependant elle fait partie des attentes à la fin du projet.

Description de la fonction « Consultation »

Cette fonction permettra de consulter l’état de nos applications, afin de connaître celle qui sont en cours de fonctionnement, celle à l’arrêt au minimum. Les prochains arrêts pourront être indiqués. Pour la consultation des performances du serveur dédié l’affichage sera limité aux administrateurs, les informations disponibles seront le pourcentage d’utilisation de la RAM, du CPU, du stockage...

d) Définition de la fonction « Maintenance & Supervision »



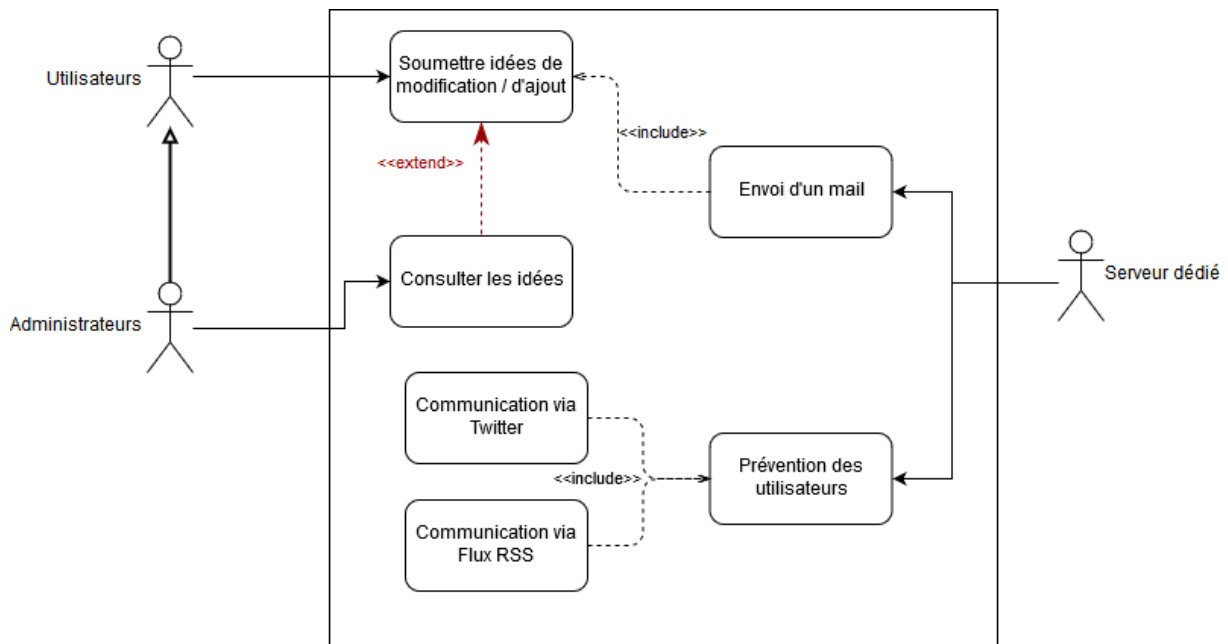
Identification de la fonction « Maintenance & Supervision »

La fonction « Maintenance & Supervision » est la partie la plus importante du projet, ce qui en fait une fonction primordiale, puisque c'est le cœur même de l'application Web. La partie « Maintenance » permet de gérer le statut des applications de notre parc. Cette dernière est liée à la partie « Supervision », puisqu'il sera possible de configurer le redémarrage automatique ou un arrêt programmé d'une application.

Description de la fonction « Maintenance & Supervision »

La fonction « Maintenance & Supervision » pourra rendre possible de lancer et de stopper une application de notre parc mais aussi mettre à jour celle-ci, si besoin. Sans oublier le fait de pouvoir configurer son redémarrage automatique en cas de crash, ou autre raison mais aussi un arrêt programmé pour une mise à jour prochaine, par exemple.

e) Définition de la fonction « Communication »



Identification de la fonction « Communication »

La fonction « Communication » sera une fonction facultative, elle permet dans différents cas de communiquer, soit via un mail ou par les réseaux sociaux. Ce n'est pas une fonction importante dans le fonctionnement de l'application Web.

Description de la fonction « Communication »

Si cette fonction n'est pas importante dans le fonctionnement de l'application Web, d'un point de vue utilisateur, elle peut l'être. Elle permet notamment à nos utilisateurs de communiquer des idées de développement mais aussi d'ajout ou modification d'une application, cette communication se fera par un mail qui sera communiqué aux administrateurs de l'application Web. Quant à nos administrateurs, ils pourront donc consulter ces idées via l'application Web. Le serveur dédié pourra prévenir tous nos utilisateurs confondus via les réseaux sociaux (exemple Twitter) et/ou via un flux RSS de tout changement d'état d'une application ou de tout arrêt programmé.

7. Conditions de fonctionnement

a) Performances

Du point de vue de l'utilisateur, il faut que l'interface de l'application Web soit toujours disponible. Il doit donc toujours se passer quelque chose à l'appui ou à la modification d'une configuration.

Au niveau de la gestion de l'environnement, le serveur dédié devra toujours assez de capacité pour pouvoir tourner l'application Web.

b) Capacités

Les limites de stockage et de performances seront fixées par le serveur dédié utilisé, ce serveur ne servira qu'à ce projet, il est donc possible d'utiliser le maximum des performances du serveur. Étant limité par les performances d'un serveur, cela signifie qu'il n'y aura pas un nombre indéterminable d'applications tournant en même temps sur le serveur, il faudra donc couper certaines applications pour en lancer d'autre.

c) Modes de fonctionnement

Le serveur dédié et l'application Web doivent continuer à fonctionner, même quand toutes les applications de notre parc sont éteintes.

d) Contrôlabilité

Les informations de statuts pour chaque application, au moment d'un lancement, redémarrage ou l'arrêt doivent être consultable depuis l'application Web. Il sera possible en cas de besoin, d'accéder au fichier de log de chaque application (si le chemin est précisé dans la configuration de l'application).

e) Sécurité

Deux types d'utilisateurs seront présent au minimum sur notre application Web. Pour les deux types d'utilisateurs devront se connecter via un identifiant et un mot de passe unique à chaque utilisateur.

f) Conformité aux standards

La création de l'application Web, sera sujet aux standards W3C ainsi qu'à la création d'un site qui sera avec un « responsive design », afin de permettre l'affichage sur toutes les plateformes.

II. Plan de développement

1. Découpage du projet en tâches

a) Tâche « Cahier de spécifications »

Description de la tâche

Il s'agit de réaliser un planning pour le projet ainsi qu'un cahier de spécifications.

Livrables

À l'issue de cette tâche, un document contenant ces deux informations sera remis.

Estimation de charge

La tâche est estimée à 45 jours car on ne travaille pas sur le projet à temps plein.

Contraintes temporelles

La tâche doit impérativement être terminée avant le 10/11/2018.

b) Tâche « Cahier d'analyse »

Description de la tâche

Il s'agit de réaliser une analyse UML/SysML détaillée du projet, permettant de supprimer des doutes et d'aider à l'implémentation du projet.

Cycle de vie

Une première version sera rendue pour le mois de décembre 2018, le 16/12/2018. Au fur et à mesure de l'implémentation de l'analyse, cette dernière pourra être corrigée pour mieux refléter la réalité et les problèmes rencontrés pendant le développement et l'implémentation du système. Une version finale sera rendue à la fin du PFE.

Livrables

Un cahier contenant plusieurs diagrammes sera rendu aux deux dates.

Estimation de charge

En prenant en compte les contraintes de temps de travail sur le projet, la tâche est estimée à 40 jours.

Contraintes temporelles

Première version doit être rendue pour le 16 décembre 2018. Une seconde version qui sera la version finale, sera rendue à la fin du PFE, courant du mois de février.

c) Tâche « Étude & Mise en place »

Description de la tâche

Il s'agit de trouver des solutions techniques à la réalisation du projet et le système d'exploitation adéquat.

Livrables

Les résultats des expériences, ainsi que celles des tests des solutions constitueront les livrables de cette tâche. L'installation du système d'exploitation choisi sera réalisée sur le serveur dédié.

Estimation de charge

La tâche est estimée à 28 jours. Des rendez-vous réguliers avec l'encadrant du projet pour convenir de la solution la plus adéquate.

d) Tâche « Réalisation de l'application Web »

Description de la tâche

Cette tâche sera constituée d'une partie prise en main du langage choisi et des solutions sélectionnés au cours de la tâche précédente, si besoin. Le développement de l'application se fera par la suite, ainsi que le traducteur pour faire comprendre les messages, provenant du site Web, aux applications du parc.

Cycle de vie

Une première version sera disponible le plus tôt possible puis plusieurs améliorations seront envisageable suite aux retours des utilisateurs et des encadrants.

Livrables

Le site Web et le traducteur devront être fonctionnels.

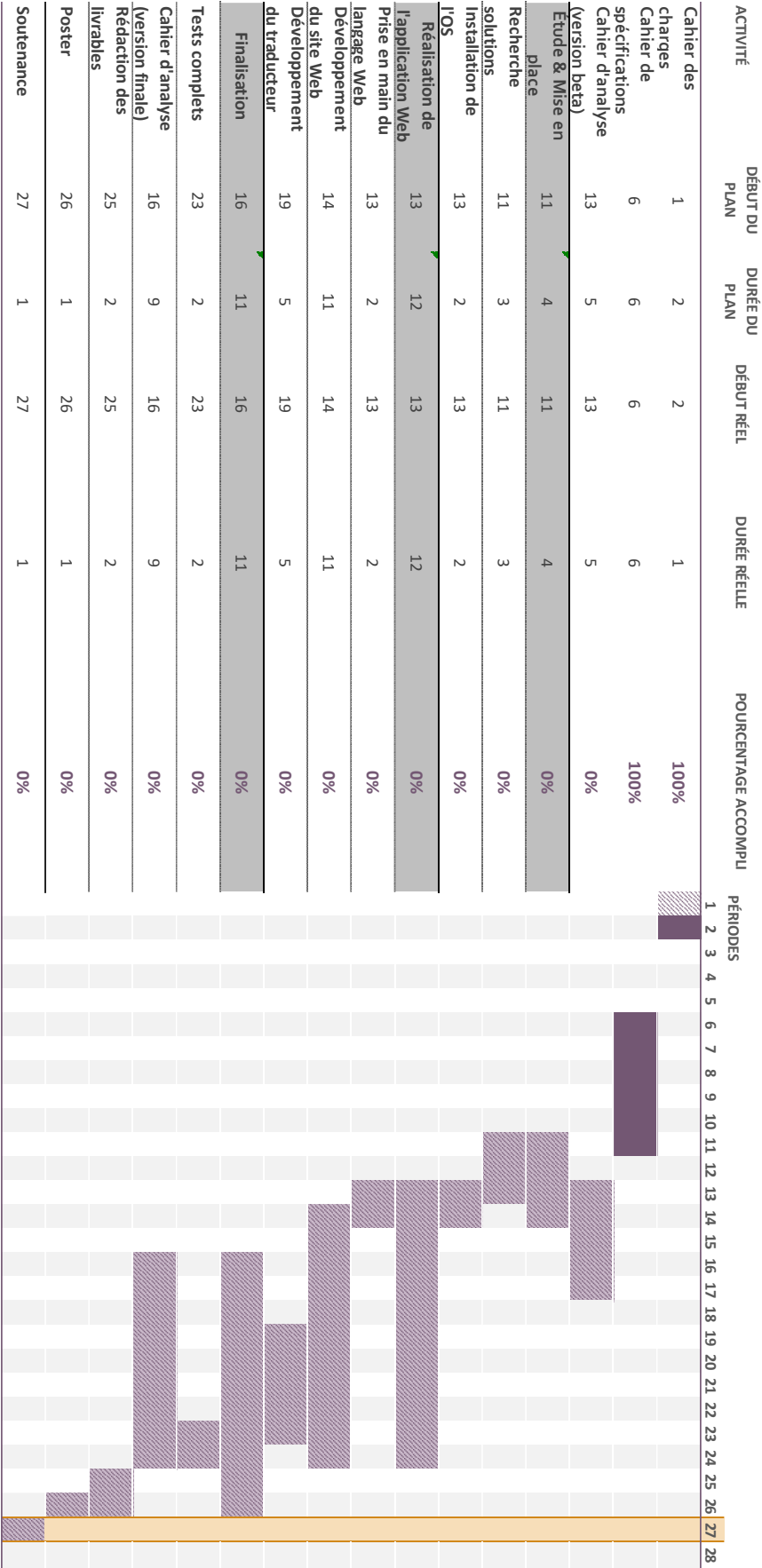
Estimation de charge

La tâche est estimée à 84 jours comprenant la prise en main et le développement.

Sélectionnez une période à mettre en évidence à droite. Une légende décrivant le graphique suit.

Période à mettre en évidence : 27

- Durée du plan
- Début réel
- % accompli
- Réel (au-delà du plan)
- % accompli (au-delà du plan)



2. Planning